SRL: Providing a Bidirectional Abstraction for Unidirectional Ad Hoc Networks

Venugopalan Ramasubramanian*
Department of Computer Science
Cornell University
Ithaca, NY 14850
ramasv@cs.cornell.edu

Daniel Mossé†
Department of Computer Science
University of Pittsburgh
Pittsburgh, PA
mosse@cs.pitt.edu

Abstract

Several routing protocols for mobile ad hoc networks work efficiently only in bidirectional networks. Unidirectional links may exist in a real network due to heterogeneity in transmission power of different nodes, noise, and other signal propagation phenomena. Using statistical analysis, we show that presence of unidirectional links severely affects the connectivity in the network. To alleviate this problem, we introduce a sub-layer called Sub Routing Layer, SRL, between the network and the MAC layers, and provides a bidirectional abstraction of the unidirectional network to the routing protocols. We present a scalable and efficient way to provide this abstraction by adaptively finding and maintaining multi-hop reverse routes to each unidirectional link. Simulations of a modified version of AODV that uses SRL to route packets in unidirectional networks show that the packet delivery increases substantially (up to 40%) compared to regular AODV.

1 Introduction

A network of mobile nodes using peer-to-peer communication and without a fixed communication infrastructure is called an *ad hoc network*. The lack of infrastructure enables quick deployment of such networks. Hence, they are very useful in disaster recovery, collaborative work, rescue operations and military surveillance. Ad hoc networks are thus likely to consist of heterogeneous hardware with different limitations in terms of power, memory, bandwidth and computation capabilities. They are also likely to be deployed in very hostile environments that significantly limit the operation of the devices. For example, an ad hoc network may be deployed in a forest fire to enable fire fighters to gather information and control the fire. Such a network would consist of sensor nodes to measure atmospheric conditions, data processing nodes to analyze the data, as well as, laptops or hand-held devices carried around by fire fighters to interface with the network. This forms an ad hoc network

*Supported in part by DARPA/AFRL-IFGA grant F30602-99-1-0532 and in part by NSF-CISE grant 9703470, with additional support from the AFRL-IFGA Information Assurance Institute, from Microsoft Research, and from the Intel Corporation.

†Supported in part by DARPA PARTS (Power-Aware Real-Time Systems) project under Contract F33615-00-C-1736 and through NSF-ITR medium grant.

with considerable heterogeneity deployed in a harsh environment that significantly restricts signal propagation in certain directions.

A fundamental problem in an ad hoc network with heterogeneous transmission power is the creation of unidirectional links in the network. For example, in Figure 1 node A is transmitting at higher power than other nodes. Node B is within the transmission range of node A while node A cannot hear node B. Thus the link $A \to B$ is unidirectional. Even when nodes are transmitting at the same power, unidirectional links may be created due to random fluctuations in signal propagation and presence of noise sources near a node that affect packet reception at that node more than another.

Recent deployments of ad hoc networks indicate a significant presense of unidirectional or asymmetric links. Based on a real-life deployment of ad hoc nodes, Ganesan et.al. report that up to 15% of the links are unidirectional even when all nodes are transmitting at the same power, and in the absence of external noise sources [9]. De Couto et. al. report that up to 30% of links have asymmetric delivery rate in an indoor deployment of wireless nodes [6]. Finally, a recent study of packet delivery performance by Zhao et. al. find that more than 10% of links have significant asymmetry in their packet delivery rates [25]. Overall, these experiments indicate that asymmetry is a fundamental problem in ad hoc networks.

The presence of unidirectional links severely affects the functionality of several routing protocols. Some routing protocols (e.g., TORA [17]) were primarily designed only for bidirectional networks and hence break down in the presence of unidirectional links. Several others (e.g., AODV [18]) continue to operate by avoiding the unidirectional links and routing data only along the bidirectional links. A few other protocols (e.g., DSR [12]) have the capability to route packets using unidirectional links. However, they employ expensive mechanisms that significantly decrease the throughput of the network. A detailed review of existing routing protocols that handle unidirectional networks in different ways is presented in Section 7.

In addition to the routing layer, unidirectional links also pose several problems at the lower layers such as data link layer and MAC layer. The failure of both RTS-CTS and ACK based schemes make MAC protocols inefficient in a unidirectional environment. Many MAC protocols also provide other services that are useful to the routing protocols. They monitor the status of the links with adjacent neighbors and hence detect a link-break or a new neighbor and report the events to the higher layer protocols. Services such as detection of link breaks or neighbor discovery may no longer be available to the routing protocols in the presence of unidirectional links.

In this paper, we present an enabling technology that will allow routing protocols primarily designed for bidirectional networks to work with unidirectional links. We propose to introduce Sub Routing Layer (SRL), a sub-layer between the network layer and MAC layer, to present a bidirectional abstraction of the network to the routing layer. SRL provides services such as reverse-route forwarding, link status sensing, and hop-by-hop reliability that amends the problems posed by unidirectional links. A dynamically adaptive variant of SRL, called DSRL, enables SRL to scale its overhead with the amount of asymmetry in the network. We do not consider the details the SRL implementation within the current protocol stack. Rather, we focus on the principles that are needed to provide such an abstraction, and on the implementation that will work with one routing protocol, namely the well-known AODV. Other implementations will be carried out in a similar manner, whereby the routing layer can take advantage of the bidirectional abstraction provided by SRL. We presented the core algorithm and a preliminary evaluation of SRL in [20] and [21]. This paper presents an elaborate description of SRL, a simulation-based evaluation of SRL across wide range of network conditions, and a statistical analysis of unidirectional networks that motivates this work.

We organize the paper in the following way. First, we present a brief overview of SRL in Section 2. In Section 3, we describe an extensive statistical analysis of topologies with unidirectional links and describe important characteristics of connectivity in asymmetric networks. We describe the services provided by SRL and detail the techniques used by SRL and DSRL to operate efficiently in Section 4. In Section 5, we present a framework to adapt on-demand routing protocols to operate over SRL. We present simulation results and discuss the performance of AODV over SRL and DSRL in Section 6. In Section 7, we describe existing techniques to handle asymmetry in ad hoc networks. We conclude the paper in Section 8.

2 Overview of SRL

We propose a solution to the problem of asymmetry by introducing a Sub Routing Layer, that offers a bidirectional abstraction of the network to the routing layer. SRL facilitates accurate identification of unidirectional links, enables routing protocols to route data packets along unidirectional links, reaching nodes that cannot be reached using only bidirectional links, and supports reliable transmission of data across unidirectional links. By providing these services, SRL enables existing routing protocols to operate efficiently in the presence of unidirectional links with little or no modification. In this section, we present a brief overview of the operating principles of SRL.

The basis of SRL is the concept of *reverse route*s for unidirectional links. In Figure 1, $B \rightarrow C \rightarrow A$ is a reverse route for the unidirectional link $A \rightarrow B$, while $E \rightarrow D \rightarrow C \rightarrow A$ forms a reverse route for unidirectional link $A \rightarrow E$. Protocols that utilize bidirectional links to send route replies and route maintenance packets (such as AODV) can now route data packets through link $A \rightarrow B$ while using the reverse route $B \rightarrow C \rightarrow A$ to send route replies or route error packets. Thus, having a reverse route, even if it is few hops long, can make a unidirectional network easily operable for bidirectional routing algorithms. While it is true that the reverse route may be longer (in number of hops and delay) than the forward route, the penalty for using these reverse routes is not critical, because only control packets (e.g., route errors and route replies) are routed through them; this increases only the route discovery latency.

SRL operates by discovering and maintaining the shortest reverse routes for unidirectional links in the network. It uses a modified version of the classical Bellman-Ford algorithm, called *reverse distributed Bellman-Ford*, to find reverse routes. Our distance vector algorithm does not suffer from the counting-to-infinity problem caused by bouncing effect, as well as the routing loop problem. SRL embodies a locally proactive protocol to disseminate the reverse route information, but the periodic updates generated by a node are restricted to a small region around that node. This enables SRL to incur low overhead and scale with the size of the network.

We also propose a dynamic variant of SRL, DSRL, which changes the size of the region around which a node's periodic updates are propagated. This dynamic adjustment of the size of the locality, enables SRL to adapt to the changing levels of asymmetry in the network. In addition to dynamic adaptation, SRL employs several other mechanisms to reduce its overhead and operate efficiently across a wide range of networking conditions. We describe the techniques used by SRL and DSRL, and the services provided by them, in great detail in Section 4.

3 Topology Analysis of Unidirectional Networks

Before presenting the details of SRL, we first study the impact of the presence of unidirectional links on network characteristics such as connectivity. Since it is difficult to perform rigid mathematical analysis and obtain useful closed form expressions, we resort to performing simulations in order to statistically analyze network topologies. We generate random topologies and analyze them for connectivity in the presence and absence of unidirectional links. Each of these topologies represents an instantaneous snapshot of the network, and the variation with time due to mobility is modeled as a series of snapshots. As we will see below, this elaborate statistical analysis of several topologies offers several valuable insights for efficient routing in unidirectional network.

3.1 Notation and Definitions

The topology of a network is a directed graph, $D = (V, E)$, where V is the set of nodes in the network and E the set of links in the network. A *link* $A \rightarrow B$ exists between two nodes A and B if B is within the transmission range of A. A link $A \rightarrow B \in E$ is *bidirectional* if $B \rightarrow A \in E$ and *unidirectional* if $B \rightarrow A \notin E$. The *reverse route* of a link $A \rightarrow B$ is the shortest directed path from B to A and the length of this shortest path is the *reverse route length* of the link. If no such path exists between B and A then the reverse route and the reverse route length are not defined. A *bidirectional* link has a reverse route length of 1 hop. We say the network is *strongly connected* when every link has a reverse route.

We categorize the links in the network based on the reverse route length. A link with reverse route length r is an *r-link*. Thus *1-link*s represent the set of all bidirectional links in the network. The *r-graph* of a network is the sub-graph consisting only of the links with reverse route of length at most r. By this definition, the ∞-*graph* of the network consists of all the links that have a reverse route. Hence, each component in the ∞-graph is strongly connected. The *1-graph* of a network includes only the bidirectional links and hence represent the routing structure used by routing protocols that route only using the bidirectional links [18].

3.2 Analytical Models

We randomly generated several topologies with parameters commonly used in simulations of routing protocols. Each topology consisted of 100 nodes placed in a square field randomly with a uniform probability distribution. We varied the area of the square field by changing the density of the nodes from 30 nodes/km^2 to 100 nodes/km^2 in steps of 10. We also performed similar trials with rectangular fields. The results observed were qualitatively similar to those presented here.

We used three different models to create unidirectional links in the topologies.

1. **P-model**: We simulated unidirectional topologies created by random fluctuations in signal propagation. We initially assigned a nominal transmission range of 220m to all the nodes and then probabilistically converted links to unidirectional. We varied the probability of unidirectional links from 0 to 0.3 in steps of 0.05.

2. **N-model**: We simulated unidirectional topologies created by the presence of noise sources. We distributed the noise sources throughout the network at random (uniform distribution). Each noise source

had a power 50 times less than the nodes in the network (recall that the nominal transmission range of regular nodes is 220m). We varied the number of noise sources from 0 to 30 in steps of 5.

3. **D-model**: We used the D-model to generate unidirectional topologies with diversity in transmission ranges for each node. The *diversity* of a topology is the difference between the maximum and the minimum transmission ranges of the nodes in the network. According to the D model, we assign to each node a transmission range picked randomly (uniform distribution) from a set of transmission ranges in the interval $[N - \frac{D}{2}, N + \frac{D}{2}]$, where N is the nominal range (typically set to 220m). In order to simulate radios with continuous and discrete steps of transmission ranges, we varied the granularity in picking the transmission ranges from 1m to 60m. For example, for a diversity of 80m and a granularity of 40m, we would randomly pick a transmission range from the set, 180m, 220m, and 260m. We varied the value of the diversity between 0m and 320m in steps of 40m; due to similarity of results, we present here only the results for the granularity 40m.

For each set of parameters described above we randomly generated 500 topologies and analyzed them statistically. The following sub-sections describe the observations made from this analysis. In the graphs that we present, each data point corresponds to one experiment with 500 trials of random topologies. The error bars plotted in the graph show the 99% confidence interval of the values. We repeated these experiment with 50, 200, and 400 nodes for the same values of density but, interestingly, we found that the average connectivity is approximately the same as that for 100 nodes, at each density.

3.3 Link Distribution Statistics

The graphs in Figures 2(a) and 2(b) show the percentage of r-links in the network for different values of r in the N-model and D-model, respectively, for topologies with density of 50 nodes/km^2. The Y-axis shows the percentage contribution of each r-link (for $r = 1, 2, 3$) in the total number of links with reverse routes in the topology. The contribution of r-links for $r > 4$ is negligible and hence not shown in the graphs.

We make two important observations from these graphs. First, a significant percentage of links in these topologies are unidirectional. The graph in Figure 2(a) shows an increase in the percentage of unidirectional links from 0% to 15% as the number of noise sources varies from 0 to 30, and the graph in Figure 2(b) shows an increase in the percentage of unidirectional links from 0% to 33% as the diversity increases from 0m to 320m. Second, the contribution of r-links decreases sharply with increasing r. These graphs indicate that up to 97% of the links have very short (i.e., ≤ 3) reverse route length.

3.4 Connectivity Statistics

We next examine the connectivity characteristics of unidirectional networks. In this analysis, we use the size of the largest connected component as an indication of connectivity in the network.

The graphs in Figures 3(a), 3(b), and 3(c) show the average size of the largest connected component of the 1-graph at different densities for the P-model, N-model, and D-model, respectively. The values plotted are normalized to the average size of the largest connected component in the ∞-graph. These graphs indicate a general decrease in the size of the largest component in the 1-graph. For example, at a density of 50 nodes/km^2, the average size of the largest component in the 1-graph drops to 93% in both the N-model and the D-model. The decrease in connectivity is even greater at higher densities in the N-model (because the noise sources are also denser), and at lower densities in the D-model (15% decrease at 40 nodes/km^2) and the P-model.

The graphs in Figure 4(a) and Figure 4(b) show the average size of the largest connected component for different values of r in the N-model and D-model respectively, for a density of 50 nodes/km^2. The sizes shown in the figure are normalized with respect to the size of the maximum component in the absence of unidirectional links. There is a significant increase in the size of the largest component of the 2-graph compared to the 1-graph in both the N-model and the P-model. The connectivity of r-graph continues to improve as r increases. However, there is diminishing returns with increasing values of r, that is, the average improvement in connectivity decreases as r increases.

The frequency distribution of the size of the largest component provides a deeper understanding of the impact of unidirectional links on connectivity. The graph in Figures 5(a), 5(b), and 5(c) plot the frequency distribution (histogram) of the size of the largest component in the 1-, 2-, and 3-graph (respectively) of the 500 randomly generated topologies with diversity 200m and density 50 nodes/km^2. The x-axis gives the size of the largest component, and the y-axis shows the number of topologies with the corresponding size for the largest component.

The graph in Figure 5(a) indicates heavy clustering of values towards the right implying that the size of the largest component is very high (> 90) mostly. The discerning feature of the graph is the heavy tail of the distribution growing to very low sizes for the largest component. This suggests that occasionally the unidirectional links could drastically reduce the connectivity of the network. The frequency distributions for the N-model and P-model (not included due to similarity of results) also show a heavy tail distribution as well. Examining Figures 5(a), 5(b), and 5(c) we can see that the values shift to the right as r increases, showing that the connectivity improves significantly when 2-links are included. The tail in the distribution is lighter indicating that the standard deviation of the size of the largest component gets smaller as r increases. Nevertheless, the heavy-tail nature persists.

3.5 Summary of the Observations

The statistical analysis of the topologies generated indicate that the presence of unidirectional links significantly affects the connectivity of the 1-graph, and that most of the unidirectional links in a random graph have short reverse routes. The connectivity follows a heavy tail distribution and can suddenly change from good to worse in real-life due to mobility. However, the connectivity can be improved by including unidirectional links and their corresponding reverse links in routing. The heavy-tail nature suggests that an efficient routing protocol should employ unidirectional links only when really required to use them.

4 Sub Routing Layer (SRL)

The Sub Routing Layer enhances the connectivity in the presence of unidirectional links by continuously discovering and maintaining reverse routes to the unidirectional links in the network. In this section, we describe the core algorithm, the *reverse distributed Bellman-Ford* algorithm, employed by SRL to track the reverse route information. We explain in detail the locally proactive protocol that implements this algorithm, as well as several optimizations to enhance the efficiency of this protocol. Finally, we describe the services provided by SRL, which enable routing protocols to operate in asymmetric networks. But first, we define several terms that are used in this paper.

4.1 Nomenclature

Whenever a node B is in the transmitting range of node A, a link exists from A to B and it will be indicated as $A \rightarrow B$. A link is said to be bidirectional if both $A \rightarrow B$ and $B \rightarrow A$ hold. Whenever $A \rightarrow B$, the node A is said to be an *in-neighbor* of node B and B is said to be an *out-neighbor* of node A. The notation $A \rightsquigarrow_n B$ is used to indicate that a path of length n hops exists from A to B in the unidirectional network. For a link $A \rightarrow B$, the shortest route[1] through which B can reach its in-neighbor A is called the *reverse route*. For example, in Figure 1 the path $B \rightarrow C \rightarrow A$ forms the reverse route for the link $A \rightarrow B$. If a link is bidirectional then the reverse route is of length 1.

Informally, the *locality radius* sets the size of the local region called *locality* about which reverse route information is collected. For a given node A, every node that A can reach by using a path of length equal to or less than the locality radius would be included in the locality of A. Formally, $locality_r(A) = \{X | A \rightsquigarrow_i X \text{ and } 1 \leq i \leq r\}$, where r is the locality radius. SRL attempts to find reverse routes only within the locality. As a result, the locality radius also bounds the maximum length of a reverse route found. For example, locality radius 1 means the reverse routes found are all of one-hop length, that is, only bidirectional links are utilized.

4.2 Reverse Distributed Bellman-Ford Algorithm

SRL employs a distance vector algorithm to gather reverse route information. We chose a distance-vector algorithm because it needs to exchange an order of magnitude less information than a link-state algorithm. In the worst case, a distance vector algorithm exchanges at each node $O(n)$ routing information, instead of $O(n^2)$ for a link-state algorithm, where n is the number of nodes within the locality.

The Distributed Bellman-Ford algorithm is a well-known distance-vector algorithm to obtain the shortest routes between pairs of nodes in a bidirectional network. This algorithm is extremely advantageous because it works asynchronously and is guaranteed to converge if the network is not partitioned and remains unchanged for sufficient time. According to this algorithm, each node broadcasts its distance to every other node in the network. Whenever a node receives a new update from a neighbor, it re-calculates its minimum distances. For example, let $A \rightarrow B$ and let $B \rightsquigarrow_n C$. If the current known minimum distance from A to C is more than $n+1$ hops, then a new shortest path from A to C of distance $n+1$, through B, is discovered in the next round. However, this algorithm is difficult to implement in the presence of unidirectional links. In the just described case, if $A \rightarrow B$ but not $B \rightarrow A$, then A would have no way of knowing that B can reach C in n hops. Thus, A will never be able to discover the $n+1$ hop path to C.

Based on the above algorithm, we developed the *Reverse Distributed Bellman-Ford* algorithm. As the name suggests, the difference between the two algorithms is that in our algorithm the direction of routes is reversed,

[1] We use hops as the metric in this paper, although there is no restriction on how to measure distance and route lengths. Another metric of interest might be, for example, the energy consumed in the route.

that is, each node aims at finding the shortest distance *from* other nodes to itself rather than from itself *to* other nodes and our algorithm works in the presence of unidirectional links. Thus, in the case described above, if $C \rightsquigarrow_n A$, A sends a broadcast message saying that C can reach it in n hops. Since $A \rightarrow B$, B discovers that C can reach B through A in $n+1$ hops. If at B, the previous known route from C is longer than $n+1$ hops, B can now update the new $n+1$ hop route from C.

Consider the scenario in Figure 6. B knows that $A \rightarrow B$ from the first message it receives from A, but B does not know the reverse route (i.e., the route from itself to A) yet. In the second round, B neighbor-casts (i.e., broadcasts with a time-to-live of 1 hop, i.e., TTL = 1) the update message to C indicating that A can reach B in one hop. The message from C in the third round would carry two updates: $B \rightsquigarrow_1 C$; $A \rightsquigarrow_2 C$. Similarly, in the fourth round A sends out a message with the following updates: $C \rightsquigarrow_1 A$; $B \rightsquigarrow_2 A$. When B hears this update from A, the information cycle is complete and B discovers the 2-hop reverse route to A. A precise definition of this algorithm is given below.

- At periodic intervals, each node broadcasts to all its out-neighbors an update message containing the shortest paths of its knowledge from other nodes within the locality to itself.
- Whenever an update is received at node B from in-neighbor A, the following information is extracted.
  - The reverse route from B to A; this information is obtained from the entry for the route from B to A in the update received, if there exists such a route within the locality.
  - If the currently known route from node C to node B ($C \rightsquigarrow_i B$) is longer than the route from node C to node A followed by $A \rightarrow B$ (i.e., $C \rightsquigarrow_j A$ and $i > j+1$), the newly found shorter route from node C to node B through node A is recorded. Note that the path from C to A is obtained from the update message.

In addition to the length of the shortest route from each node in the locality, the address of the first hop in the shortest route is also included in the update message. Including the first-hop information provides two benefits. First, it enables each node compute the reverse route to its in-neighbors. Since B knows (see Figure 6) that the first hop in the reverse route from B to A is C, and the first hop from C to A is A, it can construct the reverse route $B \rightarrow C \rightarrow A$ easily. Second, as in [4] and [15], the first-hop information enables our algorithm to prevent the counting-to-infinity problem and to avoid routing loops that affect classical distance vector algorithms. Each node uses the first-hop information to detect routing loops and invalidates such routes from its computations.

4.3 SRL Update Protocol

SRL employs a locally proactive protocol to exchange the distance vector updates among the nodes within the locality. Once every update_interval, each node neighbor-casts an update packet carrying the reverse route information as described earlier. The periodic update packets also play the dual role of "hello" beacons used to detect link status. When a node B receives an update packet from another node A, node B detects the link $A \rightarrow B$ and adds A to its list of in-neighbors. Similarly, when a node B fails to receive at least *update_loss* consecutive update packets from an in-neighbor A, it declares the link $A \rightarrow B$ broken and recomputes all the reverse routes that use this link.

A large amount of reverse route information that is being neighbor-cast at each node is redundant. For example, once node A learns of the new route from node B, there is no need for node C to keep broadcasting the shortest route from B. Node C needs to broadcast this information only if the distance from B has changed to a new value or when the first hop of the route has changed. We optimize the operation of SRL by letting each node broadcast only the changes in the route information rather than the complete information in each round. Since some of these broadcast messages could be lost because of collisions, we broadcast each change in route information twice[2].

There are cases in which it is not sufficient to incrementally broadcast the changes. A node might have moved into a new locality and is in need of route information of its new in-neighbors. In a static or slow moving environment[3], that the node might not be able to get the needed reverse routing information if the new in-neighbors broadcast the values only when they have changed. Thus, our algorithm forces each node to neighbor-cast all the routes to itself from different nodes in the locality infrequently, once every *complete_update_interval* This frequency must be set lower than the update frequency but adequately high enough for nodes to discover the new locality within a short time of moving to the new location.

The locally proactive protocol entails each node to neighbor-cast one of three kinds of update packets during each update interval: (a) infrequent complete updates with information of the order of number of nodes in the locality once every complete_update_interval, (b) incremental update packets containing only the latest changes to the same information once every update_interval, and (c) an empty update packet also called *hello_packet* to enable link status sensing in the absence of any changed routing information during an update_interval.

4.4 Dynamic SRL

The routing algorithm described earlier stipulates that the distance vector updates for each node are exchanged up to a distance of the locality_radius. Choosing an appropriate value for the locality_radius is a non-trivial task. A small value of 2 to 3 hops could be chosen for the radius, since the statistical analysis in Section 3 indicates that 97% of the unidirectional links have reverse route of length less than 3 hops. However, the heavy-tail nature suggests that connectivity could be significantly degraded some time if unidirectional links with longer reverse route lengths are not used. Using a larger value for the radius would entail very large overhead for the locally proactive protocol, since the number of nodes in a locality increases polynomially with the radius ($O(r^2)$ in a two dimensional topology).

In order to reduce the overhead of choosing a larger radius, we propose a dynamic variant of SRL, called DSRL, that adjusts the locality_radius of each node independently. DSRL continuously monitors the reverse route lengths required to reach the in-neighbors of each node, and adapts the value of the locality_radius of that node accordingly. Thus each node has a different locality_radius independent of other nodes. The locality_radius of each node is exchanged as part of the distance-vector updates. A node B includes the update of another node A in its periodic packet if A can reach B within r hops, where r is the locality_radius of A.

At each node, the DSRL adaptation algorithm is periodically invoked to adjust the radius before an update

[2] We realize that this is, by no means, optimal; we are studying ways of making this practical approach more efficient.

[3] In a fast moving network, the frequent link-breaks would trigger incremental updates from neighboring nodes. The new node can then apprise itself from these messages.

packet is neighbor-cast. The adaptation algorithm adjusts the value of the radius based on two considerations: local considerations, that is, the reverse route lengths required to reach the in-neighbors, and global considerations, that is, the radius of other nodes in the locality. It finally sets the new radius to the maximum of the radius chosen based on the two considerations. Nodes in the locality exchange incremental updates to propagate changes in the radius as part of the locally proactive protocol.

The DSRL adaptation algorithm computes a new radius based on local considerations as follows: If more than *dsrl_tolerance_threshold* percentage of the asymmetric links do not have reverse routes, the new radius is set to *dsrl_max_radius*. Otherwise, the new radius is set to the maximum reverse route length of the asymmetric links at that node. The DSRL adaptation algorithm also computes a new radius based on global considerations. If a node has a radius of r, then a second node at a distance d from the first node must have a radius of at least $r - d$, in order to ensure that the first node can obtain a reverse route through the second node. For example in Figure 6, if node B has a radius of 3, then node C must have a radius of at least 2 and node A at least 1. Thus the new radius based on global consideration is the maximum value of the difference between the radius and the distance of all nodes in the locality.

The dynamic adaptation algorithm of DSRL ensures that an appropriate radius is selected at each node to find the reverse routes of asymmetric links in the network. The dsrl_tolerance_threshold is used to ignore some asymmetric links from the consideration as the statistical analysis indicates that there are some asymmetric links with no reverse routes in the network. Overall, DSRL helps to decrease the total overhead significantly and allows SRL to use a moderate value of dsrl_max_radius without incurring excessive overhead.

4.5 SRL Interfaces and Services

The abstraction of bi-directional network as provided by SRL is not completely transparent, that is, all packets generated by the routing layer need not necessarily pass through the sub-routing layer. Efficiency considerations led us to allow the bypassing of SRL, permitting the routing protocols to communicate directly with the MAC layer. We envision that only the control packets are routed using the reverse routes, thus the data packets can be directly transmitted without incurring any delay because of an extra layer. For this reason, we chose not to provide a totally transparent interface to the routing protocols. Further, we believe that the routing protocols can be adapted more efficiently if they use the services of SRL intelligently and only upon necessity. For example, a totally unaware routing protocol that knows that there is a direct route $A \rightarrow B$ might assume that the reverse route is a fast, direct-hop route; this would cause node B to start routing packets (by mistake) through the longer reverse routes. For these reasons, the routing protocols are expected to be aware of the unidirectional nature of the network, to choose the proper routes without misconceptions on the length of the routes.

Reverse Route Forwarding

The fundamental bidirectionality abstraction of SRL is provided by a service through reverse routing or sub-routing. For example, an on-demand routing protocol can use the reverse route to send route replies back to the sender of the request. Without SRL, if any of the forward route links are unidirectional, the protocol would fail; with SRL, the route reply packet is sent to SRL for reverse routing. SRL looks up the reverse route information it stores, finds a reverse route, and forwards the packet using this reverse route. SRL appends an

IP option containing the reverse route so that the packet can be source-routed along the reverse route. Since the locality radius is typically a small number, the length of the source-route IP option remains small. The SRL-routed packet is then delivered to the routing protocol at its destination. Source routing is required because for a unidirectional link $A \rightarrow B$, only node B can compute the entire reverse route; intermediate nodes of the reverse route may not know the next node in the reverse route of the unidirectional link.

Reliable Packet Delivery

In a unidirectional environment, the MAC protocol is unable to provide any guarantee for delivery since both RTS-CTS schemes and hop-by-hop ACK-based schemes cannot be implemented at the MAC layer. SRL provides interfaces to send packets reliably both along a unidirectional link as well as along a reverse route. SRL provides reliability service by implementing an ACK mechanism. Sequence numbers are used to uniquely identify messages and avoid duplicate message delivery.

Each packet transmitted by SRL over a unidirectional link or reverse route is tagged with a sequence number using an IP option header. Sequence numbers are maintained for each pair of nodes in the locality. The sequence number is set to 0 whenever nodes discover each other in the locality. The sequence numbers are then incremented upon each message transfer.

The destination of the packet would send an acknowledgment with the sequence number back to the source node. If the packet is sent along the direct link then the acknowledgment is sent along the reverse route of the link. If the packet traversed the reverse route in its forward journey, the acknowledgment is sent through the direct link. The time-out value for receiving an acknowledgment is set proportional to the number of links traversed by the packet and the acknowledgment. Thus if a packet is sent along a direct link $A \rightarrow B$ and this link has an r hop reverse route then the time-out is proportional to $r + 1$. Similarly, if the packet is sent along a reverse route of length r then the time-out is proportional to $r + 1$. The time-out for a packet sent on a direct link is specified by the *perhop_ack_timeout* parameter.

SRL attempts to re-transmit each packet (upon time-out) as many times as specified by the *num_ack_retries* parameter. After that, if no acknowledgment is received then a packet-drop event is raised to the routing protocol. This event can be used to detect link breaks in the network. SRL currently uses a simple retransmission mechanism to decrease packet loss and detect link breaks. Sophisticated mechanisms, such as exponential back-off and multiple outstanding packets, could reduce network congestion and improve throughput. Detailed study of these techniques is beyond the scope of this paper, but we intend to address it in future.

Link Discovery

In addition to the routing services, SRL provides additional services that are sometimes provided by the MAC layer in a bidirectional environment. Since SRL has an inherent neighbor discovery mechanism, it can export this mechanism to higher layers. For that, our implementation of SRL raises new *in-neighbor found* and *in-neighbor lost* events. Routing protocols may use these events to initiate route repairs. Many routing protocols designed for bidirectional networks use packet drops at MAC layer to detect link breaks. SRL can be made to do the same by using reliable send interfaces. Alternatively, SRL can imitate this behavior by raising a packet drop event whenever it knows that the out-neighbor cannot reach back because no reverse route exists from the out-neighbor. It is possible that the out-neighbor can receive the packet and only the reverse route does not exist. This way, SRL may generate a false packet drop event. Note that, in this situation, the MAC layer can also falsely raise a packet drop event while in fact only the acknowledgment has been lost because of congestion.

5 Unidirectional Routing with SRL

Several MANET routing protocols operate efficiently when all the links are bidirectional, but their performance degrades in ad hoc networks with unidirectional links. Since SRL provides a bidirectional abstraction of the unidirectional network, these routing protocols can use SRL to work effectively even in the presence of unidirectional links. The SRL interface enables routing protocols to operate in unidirectional networks with minimal changes. In this section, we show how AODV operating over SRL can route over unidirectional links. We can also adapt other protocols in a similar manner; we use AODV just as an example.

5.1 AODV Black-List

Ad hoc On-demand Distance Vector routing protocol, AODV [22], is an on-demand routing protocol for bidirectional ad hoc networks. Whenever a packet needs to be sent to the destination node, for which a route is not known, the source broadcasts a route request packet. When the route request packet reaches the destination or an intermediate node with fresh enough route to the destination, the destination or the intermediate node sends a route reply back to the source. If such information is not readily available at an intermediate level, the intermediate node forwards the route request.

In a bidirectional network, the reply can be unicast by reversing the request path. The destination node accepts the first route reply received as the fastest route (probably shortest number of hops away from the source or the least congested path). However, in a unidirectional network the reverse path may not exist. AODV [18] proposes to solve this problem by keeping track of the unidirectional links in a *black-list*. A node sending a route reply to its neighbor expects to receive an acknowledgment, and when the node does not receive an acknowledgment, it adds that neighbor to the black-list. This node ignores future route requests received through this neighbor. In essence, the black-list gives an approximate list of unidirectional links.

To enable adaptability to network dynamics, a node deletes each entry added to the black-list after a specified lifetime. However, it is possible that, before the deletion from the black-list, a neighbor becomes close enough to hear from this node but the route requests coming from it are ignored. Further, the presence of the unidirectional link is detected only *after* an attempt is made to send route replies through it. This may be too late and sometimes prevent routes from being discovered. Moreover, with increased mobility the black-list becomes a poor approximation.

The black-list enables AODV to ignore unidirectional links and route packets only along the bidirectional links in the network. However, certain nodes could be reachable only if a few unidirectional links are included in the route. AODV would not be able to discover such routes. Also, a link in a route that is presently bidirectional may become unidirectional in future. In such cases AODV would declare that route to be broken and start a fresh route discovery, which is an expensive procedure.

5.2 AODV over SRL

We propose and study AODV on SRL, for the following reasons. By using the services of SRL, AODV obviates the necessity for a black-list. Also, because SRL maintains reverse routes, it can identify with good accuracy whether a link along which a route request is received is unidirectional (in fact, SRL with a locality radius of 1 is sufficient to do this). SRL enables AODV to build routes to nodes that require some unidirectional links to reach them. Lastly, SRL improves the efficiency of AODV because SRL avoids new route discoveries when bidirectional routes become unidirectional.

We modified AODV to route packets through the unidirectional links using SRL. Nodes send route replies, as well as route errors, through a multi-hop reverse route with SRL. Otherwise, the behavior of AODV remains the same as before. Sub-routing route replies may induce an additional delay in receiving route replies at the source, but it only affects the route discovery latency as the data packets continue to travel the same forward path or a shorter forward path. (SRL enables AODV to discover shorter forward paths using unidirectional links, which would otherwise be black-listed by AODV).

While routing along unidirectional links, the MAC protocol used may not guarantee reliability, and may not be able to detect link breaks. So, the modified AODV uses the services of the Sub Routing Layer to send packets reliably and detect link breaks. AODV uses the reliable send feature of SRL to send data packets as well as route replies and route errors. SRL generates packet-drop events whenever it fails to reliably send packets. AODV treats the packet-drop events in the same way as it treats the packet-drop events generated by the MAC layer (e.g, IEEE 802.11). Thus, SRL-AODV initiates route error messages upon notification of packet drops by SRL, not the MAC layer.

6 Simulation Study

We studied the performance of the Sub Routing Layer with respect to the overheads it imposes and amount of routes enabled by implementing it in a simulation environment. In this section, we describe the simulation environment and present results from the simulations in detail.

We simulated SRL-AODV and DSRL-AODV, an implementation of AODV with SRL and DSRL. We used an implementation of AODV as described in [18] defining the same values for the parameters. For simplicity, we chose not to implement the *gratuitous reply* and the *local error recovery* optimizations of AODV. However, we included other optimizations such as expanding ring search and route reply generation from route.

We studied the performance of SRL for each value of radius from 1 to 8 as well as a very high value set to the diameter of the network. Table 1 lists the values used for the parameters of SRL and DSRL. These values were chosen based on controlled experiments with different parameters (0.3s to 1.5s for update_interval, 1 to 3 for update_loss, 5ms to 25ms for perhop_ack_timeout, 2 to 4 for max_ack_retries, and 0% to 20% for dsrl_tolerance_threshold). For the sake of comparison, we simulated AODV with black-list operation (AODV-BL) as described in [18].

6.1 Simulation Environment

We used GloMoSim [24], a scalable packet-level simulator for wireless networks. It uses parallel discrete event simulation provided by Parsec [1], a parallel C-based simulation language.

We set the bandwidth of the physical channel to 2 MHz. For the radio-layer, we employed a two-ray path propagation model to simulate signal propagation. The nominal transmission range of this model was 220m corresponding to the WaveLan radio hardware. We used IEEE 802.11 as the MAC protocol for AODV-BL. Since AODV-BL only routes packets along bidirectional links and 802.11 uses acknowledgments and detects packet drops, it is well-suited for AODV-BL. The SRL-AODV and DSRL-AODV simulation used CSMA as the MAC layer.

We performed simulations in several different environments and configurations, but due to the similarity of the results, we present here only the results for a topology of 80 nodes randomly distributed in an area of $1300m \times 1300m$. We set the simulation to execute for 360 seconds. We set the transmission range of each node based on the D-model described in Section 3. Accordingly, each node was randomly set a value for the transmission range from the interval $[N + \frac{D}{2}, N - \frac{D}{2}]$, where N is the nominal range of 220m. We simulated with six different values of diversity, D: 0m, 80m, 160m, 240m, 280m, 300m. For each set of parameters, we ran 50 trials of simulations with different seeds for random number generation. The need for several trials is justified by the long-tail distribution of connectivity, as shown in Section 3.

We used the Constant Bit-Rate generator, CBR, as the application. In each trial, CBR sent data packets between 20 randomly chosen sources and destinations. For each source-destination pair, data traffic originated at a randomly chosen time between 50s and 150s and terminated at a randomly chosen time between 250s and 350s. In this duration, each source sent 200 packets with a random size between 64B and 1024B. Thus the data rate varied for different sources, while the number of packets sent in each trial remained a constant. In total, CBR generated 4000 packets in each trial at an average date rate of 1 packet per second.

We used the random-waypoint mobility model to simulate nodes in motion: each node chooses a random point in the environment as the next destination and a random speed (between a maximum and a minimum value), and moves towards that destination with the chosen speed. After reaching the destination, the node waits for a specific *pause time* and restarts the motion again. We executed the simulations for two scenarios: *low-speed or pedestrian mobility* with speeds between 1m/s and 2m/s and *high-speed or city vehicular mobility* with speeds between 10m/s and 12m/s. For both the scenarios, we varied the pause time from 0s to 360s in steps of 60s. A pause time of 360s corresponds to a static environment.

6.2 Connectivity

The principal goal of SRL is to improve the connectivity of the network by using unidirectional links for routing. The improvement in connectivity obtained by SRL and DSRL can be studied by looking at the number of data packets sent by the source at the routing layer. This number represents the ability of the routing protocols to find routes to the destination, since AODV sends data packets only after routes are found.

The Figure 7 shows the number of data packets sent by SRL-AODV in each of the 50 trials, for a diversity of 280m and when all nodes are stationary. The graph cumulatively plots the increase in the number of data packets sent when the radius of SRL is increased. The graph highlights the heavy-tail distribution of connectivity in the presence of unidirectional links. In 25 out of 50 trials, using SRL with radius higher than 1 provides no improvement in the number of routes found. However in some trials, using SRL with radius 2 or higher improves the number of data packets sent by more than 50%.

The graph in the Figure 8(a) shows the average number of data packets sent by SRL-AODV, for different values of radius and diversity when all nodes are stationary. This graph shows that using unidirectional links for routing provides a significant improvement in network connectivity. The improvement in number of data packets sent slowly increases as the diversity of the network increases. For example when diversity is high, SRL-AODV with radius 2 provides about 10% average improvement over SRL-AODV with radius 1, which only uses bidirectional links. Increasing the radius of SRL beyond 2 also increases the number of data packets sent, but the average improvement gets smaller with increasing radius.

The Figure 8(b) shows the average number of data packets sent by SRL-AODV and DSRL-AODV in comparison to AODV-BL for different values of diversity, when all nodes are stationary. This graph shows that the number of data packets sent by DSRL-AODV, which dynamically adjusts the radius depending on the extent of asymmetry in the network, is comparable to SRL-AODV with radius 8 for all diversity scenarios simulated. The number of data packets sent by AODV-BL is marginally lower compared to SRL-1, because AODV-BL initially incurs several wasted route discovery attempts before identifying the unidirectional links.

In addition to discovering new routes to destinations by leveraging unidirectional links, SRL also finds shorter routes through unidirectional links for nodes with symmetric routes. Figures 8(c) and 8(d) show the average length of the routes found by SRL-AODV with different values of radius, DSRL-AODV, and AODV-BL. As expected, SRL-AODV with radius 1 and AODV-BL have the same average route length, as they both only use bidirectional links. However, increasing the radius enables SRL-AODV to find shorter routes to transmit data packets.

The Figure 8(e) shows the average number of data packets sent by SRL-AODV and DSRL-AODV in comparison to AODV-BL in the high-speed scenario, for a diversity of 280m. This graph helps us make the following observations. First, the number of data packets sent by AODV-BL significantly decreases as the pause time decreases, because the black-list scheme is unable to accurately identify unidirectional links in the presence of mobility. SRL-1 provides a significantly better performance (up to 35% improvement in number of data packets sent) compared to AODV-BL in the presence of mobility. Second, the number of data packets sent by DSRL is better than or comparable to SRL-8. Finally, SRL-8 sends more data packets on average than SRL-1. However, the improvement in number of data packets sent decreases with increasing mobility. This is because, SRL takes greater time to find longer reverse routes, and therefore unidirectional links become less utilized when nodes are in constant motion. Having a higher update frequency would alleviate this problem to a certain extent, but would increase the overhead.

The graph in the Figure 8(f) shows the average number of data packets sent by SRL-AODV, DSRL-AODV, and AODV-BL in the low-speed scenario, for a diversity of 280m. The observations made for the high-speed scenario are also true for the low-speed scenario, although the improvement in number of data packets sent by SRL-1 over AODV-BL is lower in the low-speed scenario.

6.3 Overhead

In this section, we present the overhead incurred by SRL-AODV and DSRL-AODV, while providing improved connectivity in the presence of unidirectional links. We first discuss the overhead incurred by the periodic update protocol of SRL and DSRL. The total number of periodic packets sent during each trial is a constant[4] 720 per node. Hence, we measure the periodic overhead by the average size of a periodic packet. The size of the periodic packet also signifies the extant of congestion due to the periodic update protocol.

The graph in the Figure 9(a) plots the average size of a periodic packet (including the MAC layer header) generated by SRL and DSRL for different values of radius and diversity, when all nodes are stationary. The average size of an SRL update packet depends on the number of nodes in the locality. Hence, it increases as the radius of SRL increases. However, it varies little with diversity because, the number of nodes in the locality does not vary significantly with diversity. On the other hand, the average size of a DSRL update packet increases gradually as the diversity in the network increases. DSRL maintains a small radius when diversity is less, incurring a low overhead, but dynamically adapts to a larger radius as the diversity increases, incurring greater overhead.

The Figure 9(b) shows how the average size of a periodic packet generated by SRL and DSRL varies with pause time in the high-speed scenario, for a diversity of 280m. The average size of a periodic packet in SRL increases with increased mobility for all radius, since more incremental update packets are sent in response to link-failures. The average size of a DSRL update packet also increases substantially with mobility. This indicates that DSRL employs larger radius as mobility increases. DSRL increases the radius in order to find reverse routes for new unidirectional links. Increased mobility causes frequent change in the unidirectional links at each node, inducing DSRL to employ larger radius.

The trends in the overhead imposed by the periodic protocol also characterize the total overhead incurred by SRL-AODV and DSRL-AODV. The Figure 9(c) plots the total number of bytes transmitted by SRL-AODV, DSRL-AODV, and AODV-BL for different values of diversity, when all nodes are stationary. We measure the total byte overhead at the MAC layer because SRL and DSRL use CSMA, while AODV-BL uses a different MAC layer protocol, IEEE 802.11. The total number of bytes transmitted by SRL-8 is higher than SRL-1 for reasons discussed earlier in this section. The overhead of SRL-1 is higher than AODV-BL because SRL being a proactive protocol incurs more overhead. The overhead of DSRL is slightly higher than SRL-1 and gradually increases with diversity.

The graph in Figure 9(d) shows how the total byte overhead varies with pause time in the high-speed scenario, for a diversity of 280m. The overhead of AODV-BL is lower than SRL-1 at low mobility but, becomes larger at high mobility. This indicates that SRL-1 incurs overhead comparable to AODV-BL for the data traffic chosen even though, SRL is a locally proactive protocol. The overhead of DSRL is comparable to SRL-1 at low mobility but, slowly increases to a value comparable to SRL-8 when nodes are in continuous motion. Overall, the overhead of DSRL is significantly lower than SRL-8, while the connectivity provided by DSRL is comparable to SRL-8. This indicates that by dynamically adapting the radius, DSRL is significantly more efficient than SRL.

[4]Note that each node transmits at least a hello packet during each update_interval, even if no incremental or complete update packet is sent.

6.4 Loss rate and Latency

In this section, we discuss the performance of SRL and DSRL in terms of the loss rate and latency. The Figure 10 shows the loss rate of SRL-AODV and DSRL-AODV in comparison to AODV-BL in the high-speed scenario, for a diversity of 280m. This graph shows that the losses incurred by SRL and DSRL is comparable to AODV-BL. At high mobility, AODV-BL has a higher loss rate than SRL due to the inaccurate determination of unidirectional links. For SRL, the loss rate of SRL-1 is lower than the loss rate of SRL with higher radius. Routes in SRL-AODV can break due to link-failure as well as failure of reverse route. Since longer reverse routes tend to break more often, the loss rate of SRL increases with radius.

The Figure 11 shows the average latency for SRL-AODV, DSRL-AODV, and AODV-BL in the high-speed scenario, for a diversity of 280m. The latencies incurred by SRL and DSRL are comparable to AODV-BL. SRL with higher radius incurs more delay than SRL-1 because, increased route breakages (due to reasons mentioned earlier) induce more route discoveries. Overall, the loss rate and latency of SRL and DSRL are as good as AODV-BL.

6.5 Summary of Simulation Results

In this section, we presented simulation results showing the performance of SRL and DSRL across a wide range of network conditions.

AODV over SRL with radius 1 routes packets only along bidirectional links, but provides a significantly better performance than AODV-BL: up to 35% improvement in number of data packets sent. However, there is clearly a trade off between the improvement achieved by SRL and the overhead imposed by it. By being locally proactive, SRL-1 is able to identify and avoid the unidirectional links more effectively, but imposes extra overhead that is independent of the data traffic. In our simulations, we show scenarios in which SRL-1 incurs lower overhead than AODV-BL, as well as, scenarios in which AODV-BL incurs lower overhead.

By using unidirectional links for routing, SRL with higher radius further improves the connectivity obtained by AODV. Even though the average improvement in connectivity is moderate (up to 15% between SRL-1 and SRL-8 for diversity 280m), simulation results indicate that many times SRL with higher radius improves the connectivity substantially (over 50%). SRL effectively compensates for the unpredictable drop in connectivity due to the heavy-tail nature.

While, SRL is able to discover more routes as its radius is increased, it also incurs greater overhead. The choice for an appropriate radius for SRL depends on the trade-off between overhead and connectivity. A smaller value of radius may be chosen for an application that can tolerate low connectivity and partitions, whereas a larger value of radius may be chosen if discovering more routes is important for the application. In essence, choosing an appropriate radius for SRL is a non-trivial task.

DSRL obviates the necessity to choose the radius in advance by adapting the radius independently at each node. DSRL provides performance comparable to that of SRL-8 in terms of data packets sent, loss rate and latency, while the overhead of DSRL slowly increases with the diversity and mobility in the network: comparable overhead to SRL-1 at low diversity and mobility, but higher overhead as diversity or mobility increases. Overall, the simulation results indicate that DSRL significantly improves connectivity in the presence of unidirectional links, incurs overhead that scales with diversity and mobility, and provides low loss rate and latency.

7 Related Work

Presence of unidirectional links renders many of the current MANET routing protocols inoperable, while some others continue to operate, routing only along the bidirectional links. Few routing protocols also route packets through unidirectional links but face many problems that decrease their efficiency. In section 5, we discussed the techniques employed by AODV to handle unidirectional links. In this section, we briefly explore the approaches taken by other manet routing protocols to handle the unidirectionality problem.

The traditional approach to this problem is presented in the charter of the UDLR (Unidirectional Link Routing) working group of IETF. The protocol specified in the internet draft [8] of UDLR working group involves encapsulating and tunneling link-layer packets as IP packets. However, we find that the specifications of this protocol is unsuitable for the purposes of ad hoc networks as the presence of a connected bidirectional network is assumed. Further, the tunneling as presented in [8] only presents a mechanism to send multi-hop acknowledgment packets at the link-layer.

In [19], the authors examine several problems associated with distance-vector routing in the presence of unidirectional links. In particular, they suggest an increase in the size of routing messages exchanged among the nodes from $O(n)$ to $O(n^2)$, where n is the number of nodes in the network. A proactive distance vector routing protocol for unidirectional networks is also proposed in the same paper. However, as acknowledged by the authors, the high overhead hampers the scalability of the protocol and makes it an inefficient solution.

Proactive protocols belonging to the link-state paradigm have also been modified to function in unidirectional networks. Such a proactive link-state approach is presented in [2]. The authors employ an explicit algorithm to discover and maintain an *inclusive cycle* for each unidirectional link, where an inclusive cycle is one formed by an unidirectional link and its reverse route. The periodic link state updates of a node are sent along the reverse routes to each of the upstream neighbors. This introduces enormous overhead in the protocol making it very inefficient for practical purposes.

DSR [12], Dynamic Source Routing, is a purely reactive routing protocol designed to work even in unidirectional networks. Routes are discovered by broadcasting route requests and route replies. Data packets are then source-routed to the destination. In bidirectional networks several optimizations can be applied to DSR and the control overhead can be reduced significantly (e.g., route caching and following the reverse route through bidirectional links for replies). However, the performance of DSR in unidirectional networks is limited by the scalability of the protocol. DSR suffers from what we call the *RREP-explosion* problem. Intermediate nodes with cached routes to the destination also broadcast route replies back to the source. Hence, several nodes might respond to the same route request, resulting in a large number of route reply broadcasts. Further, DSR relies on hop level acknowledgments in unidirectional networks for discovering route errors. This requires the discovery and maintenance of several additional reverse routes for acknowledgments at every hop. This extra control overhead results in increased congestion and severely limits the throughput of the network.

ZRP [10], Zone Routing Protocol, is a routing protocol that exploits the effectiveness of both proactive and reactive routing strategies. Each node maintains information about the topology in a small area around it called *zone*. Routing within a zone is proactive (i.e., routes are maintained for all pairs within the zone at all times) while inter-zone routing is done using a reactive protocol. A technique called *border-casting* is used to minimize the broadcast overhead of inter-zone route requests; only nodes at the border of the zone will forward/rebroadcast the RREQ. In [23], ZRP is modified to route in the presence of unidirectional links. Intra-zone routing is used to transmit route replies and route errors thus avoiding the RREP-explosion problem. Reverse routes in this protocol are gathered from the periodic packets broadcast throughout the zone.

Our framework, Sub Routing Layer (SRL), is locally proactive and hence bears superficial resemblance to ZRP. However it differs from ZRP in both the mechanism it employs, and in the goal it achieves. Unidirectional ZRP broadcasts updates to an extended zone, twice the size of a normal ZRP zone. Updates in SRL are propagated across a locality of much smaller radius. Also, optimizations in SRL's proactive routing algorithm helps it to have a considerably lower periodic control overhead than ZRP, as ZRP employs a link-state algorithm for intra-zone routing. Consequently, the overhead imposed by SRL's periodic protocol is an order of magnitude smaller than ZRP's.

IMEP [5] the Internet MANET Encapsulation Protocol, operates between the link layer and network layer providing several services including link-status sensing, broadcast reliability, and control message aggregation. IMEP can detect and monitor the occurrences of unidirectional links in the network. The service model of SRL is similar to IMEP as both the frameworks operate between the network and link layer. However, unlike SRL, IMEP does not maintain reverse routes and does not enable the use of unidirectional links for data traffic. Some of the other services provided by IMEP such as link-failure notification and neighbor discovery are also provided by SRL. SRL concerns itself only with providing support for unidirectional routing and hence provides services such as reverse-route forwarding, multi-hop acknowledgments, etc.

In [16], an alternative approach, based on tunneling, to handle the problem faced by unidirectional links is presented. Control packets and acknowledgments are tunneled back to the source enabling the routing protocols to use unidirectional links avoiding loops and explosion of acknowledgment packets. Periodic packets containing a list of neighbors are broadcast to help with the task of identifying unidirectional links. Reverse routes needed for tunneling packets are gathered and maintained by the same routing protocol as used for routing data packets. In contrast, SRL uses an algorithm that is well designed to discover and maintain reverse routes efficiently. SRL's algorithm also facilitates other services such as neighbor discovery and hop-level acknowledgments without additional overhead. In comparison, the tunneling approach delegates the entire task of maintaining reverse routes (i.e., tunnels) to the routing protocol. This places extra load on the routing protocol, which may not always be adequately suited to the task. Also, [16] advocates periodic packets to be sent by the downstream node to the upstream node along the tunnel for each unidirectional link. This approach could be improved as the advantages of a shared broadcast medium are not exploited.

In [11], the authors present an analysis of routing characteristics in the presence of unidirectional links. They study the impact of unidirectionl links created by nodes transmitting at different transmission powers. They study three variants of this model, two-power scenarios, three-power scenarios, as well as random-power scenarios. In [14], the authors presents a statistical analysis of connectivity similar to ours. However, [14] only analyzes the average behavior of connectivity, which leads it to conclude that the impact of unidirectional links on routing is insignificant. They study three variant models of heterogeneous transmission power: two-power scenatios, random-power scenarios, and scenarios created by topology control algorithms. Overall, their results agree well with our statistical analysis using the D-model.

In this paper, we provide a deeper analysis from wider scenarios, which shows that the connectivity is a heavy tail distribution, and that the presence of unidirectional links can significantly impact connectivity. We use three different models, representing three different causes of unidirectional links, to generate unidirectional topologies. On the other hand, two of the models used in [14, 11] are variants of our D-model, while the third model based on topology control used in [14] does not produce any difference in the connectivity of unidirectional and bidirectional networks. Finally, [14, 11] only discusses the connectivity of the unidirectional network vis-á-vis the bidirectional sub-graph. In this paper, we present an analysis of the connectivity of different categories of sub-graphs formed by the inclusion of unidirectional links with different reverse route lengths, which enables us to design efficient unidirectional routing protocols.

In [14], the authors also present an alternate method called *reverse path search* to avoid unidirectional links while routing. While the reverse path search is more efficient than the black-list technique employed by AODV, it does not enable routing protocols to use unidirectional routes to reach nodes that cannot be reached by only using bidirectional links. However, we do agree with [14] that there is a trade-off between the overhead and the improvement in connectivity achieved by using proactive techniques such as SRL, and that proactive techniques may impose more overhead than purely reactive strategies when the data traffic is very low.

8 Conclusions

Noise sources, heterogeneity in transmission power, and random fluctuations in signal propagation are some of the causes of the introduction of unidirectional links in ad-hoc networks. We presented a sub routing layer (SRL) that provides an ideal framework for allowing existing routing protocols to function in the presence of unidirectional links. SRL presents a bi-directional abstraction for a unidirectional environment, using a Reverse Distributed Bellman-Ford algorithm to maintain reverse routes and provide the bidirectional abstraction. We also presented DSRL, a dynamically adapting variant of SRL, which imposes a low overhead that scales with severity of unidirectionality in the network.

Statistical analysis of the topologies with unidirectional links reveals that unidirectional links significantly impact the connectivity of a network, especially because of the heavy-tail nature. Simulation results show that, SRL can enable AODV to improve the number of routes discovered by up to 15% on average by using unidirectional links for routing. SRL with radius 1 also enables accurate identification of unidirectional links, using which AODV can increase the number of data packets sent by up to 35% compared to the black-list scheme. Overall, SRL and DSRL provide significant improvement in the data delivery achieved by AODV (up to 40%) without affecting the loss rate and latency incurred by AODV.

Our current work is to develop a dynamic on-demand SRL scheme, whereby DSRL is invoked to discover reverse routes only when absolutely necessary, that is, only for unidirectional links that are necessary for establishing routes. The on-demand and dynamic SRL would enable the overhead of SRL to scale with both the data traffic, as well as, diversity and mobility, and enable SRL to efficiently tackle the heavy-tail connectivity problem.

References

[1] R. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, B. Park, and H. Song. “Parsec: A Parallel Simulation Environment for Complex Systems”. In *IEEE Computer*, Vol. 31(10), October 1998, pp. 77-85.

[2] L. Bao and J.J. Garcia-Luna-Aceves. “Link-state Routing in Networks with Unidirectional Links”. In *Proceedings of the Eighth International Conference on Computer Communications and Network, ICCCN*, 1999.

[3] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Jetcheva. “A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols”. In *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking. MobiCom 98*, Dallas, TX, October 1998.

[4] C. Cheng, R. Riley, S. Kumar, and J. Garcia-Lunes-Aceves. “A Loop-Free Extended Bellman-Ford Routing Protocol Without Bouncing Effect”, In *Proceedings of ACM Sigcomm*, pp 224-236, August 1989.

[5] M. S. Corson, S. Papademetriou, P. Papadopoulos, V. D. Park, and A. Qayyum, “An Internet MANET Encapsulation Protocol (IMEP) Specification”. *IETF Internet Draft, draft-ietf-manet-imep-spec-01.txt*, August 1999, Work in Progress.

[6] D. De Couto, D. Aguayo, B. Chambers, and R. Morris. “Performance of Multihop Wireless Networks: Shortest Path is Not Enough”, In *Proceedings of Workshop on Hot Topics in Networking (HotNets)*, Princeton NJ, October 2002.

[7] S. R. Das, C. E. Perkins, E. M. Royer, and M. K. Marina. “Performance Comparison of Two On-demand Routing Protocols for Ad hoc Networks”. In *IEEE Personal Communications Magazine special issue on Ad hoc Networking*, February 2001, pp. 16-28.

[8] E. Duros, W. Dabbous, H. Izumiyama, N. Fujii, and Y. Zhang, “A Link Layer Tunneeling Mechanism for Unidirectional Links.” *IETF Internet Draft, Network Working Group*, June 1999. Work in Progress.

[9] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, “An Empirical Study of Epidemic Algorithms in Large Scale Multihop”. *Intel Research, IRB-TR-02-003*, March, 2002

[10] Z. J. Haas and M. R. Pearlman. “The Zone Routing Protocol(ZRP) for Ad hoc Networks”. Internet draft - Mobile Ad hoc Networking (MANET) Working Group of the Internet Engineering Task Force (IETF), March 2001.

[11] Jorjeta G. Jetcheva. “Adaptive Demand-Driven Multicast Routing in Multi-Hop Wireless Ad Hoc Networks”. *PhD Thesis CMU-CS-04-126*, Carnegie Mellon University, May 2004.

[12] D. B. Johnson and D. A. Maltz. “Dynamic Source Routing in Ad-Hoc Wireless Networks”. *Mobile Computing*, edited by T. Imielinski and H. Korth, chapter 5, pp. 153-181, Kluwer, 1996.

[13] D. B. Johnson, D. A. Maltz, Y. C. Hu, and J. Jetcheva. “The Dynamic Source Routing protocol for mobile ad hoc networks”. *IETF Internet Draft, draft-ietf-manet-dsr-05.txt*, March 2001. Work in Progress.

[14] M. K. Marina and S. R. Das. “Routing Performance in the presence of Unidirectional Links in Multihop Wireless Networks”. In *Proceedings of the Third ACM Symposium on Mobile Ad hoc Networking and Computing, Mobihoc 2002*, Lausanne Switzerland, June 2002.

[15] S. Murthy and J.J. Garcia-Luna-Aceves, “An Efficient Routing Protocol for Wireless Networks”, *ACM Mobile Networks and Applications Journal, Special issue on Routing in Mobile Communication Networks*, Vol. 1, No. 2, 1996.

[16] S. Nesargi and R. Prakash. “A Tunneling Approach to Routing with Unidirectional Links in Mobile Ad-Hoc Networks”. In *Proceedings of the IEEE International Conference on Computer Communications and Networks, ICCCN 2000*, Las Vegas, Nevada, October, 2000.

[17] V. D. Park and M. S. Corson. “A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks”. In *Proceedings of the IEEE INFOCOM*, Kobe, Japan, April 1997.

[18] C. E. Perkins, E. M. Royer, and S. R. Das “Ad-Hoc On Demand Distance Vector(AODV) Routing”. *Request For Comments (RFC) 3561*, July, 2001.

[19] R. Prakash. “Unidirectional Links Prove Costly in Wireless Ad-Hoc Networks”. In *Proceedings of ACM DIAL M’99 Workshop*, Seattle, WA, August 1999, pp. 15-22.

[20] V. Ramasubramanian, R. Chandra, and D. Mossé. “Providing a bidirectional abstraction for unidirectional ad hoc networks”. In *Proceedings of the IEEE INFOCOM*, June 2002.

[21] V. Ramasubramanian and D. Mossé. “Statistical Analysis of Connectivity in Unidirectional Mobile Ad Hoc Networks”. In *Proceedings of the International Workshop on Ad Hoc Networking 2002*, Vancouver. August 18-21, 2002.

[22] E. M. Royer and C. E. Perkins. “Ad-hoc On-Demand Distance Vector Routing”. In *Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications*, February. 1999, pp. 90-100.

[23] P. Sinha, S. Krishnamurthy, and S. Dao. “Scalable Unidirectional Routing with Zone Routing Protocol (ZRP) Extensions for Mobile Ad-Hoc Networks”. In *Proceedings of the IEEE Wireless Communications and Networking Conference, WCNC 2000*, Chicago, September, 2000.

[24] X. Zeng, R. Bagrodia, and M. Gerla. “GloMoSim: A Library for Parallel Simulation of Large-scale Wireless Networks”. In *Proceedings of the Twelfth Workshop on Parallel and Distributed simulation*, May, 1998, Banff Canada, 154-161.

[25] J. Zhao and R. Govindan. “Understanding Packet Delivery Performance in Dense Wireless Sensor Networks”, In *Proceedings of Conference on Embedded Networked Sensor Systems (SenSys)*, Los Angeles CA, November 2003.

parameter	**value**
update_interval	500 ms
complete_update_interval	4500 ms
update_loss	3
perhop_ack_timeout	15 ms
num_ack_retries	3
dsrl_tolerance_threshold	15 %
max_dsrl_radius	6

Table 1: Values used in simulation for parameters of SRL and DSRL.

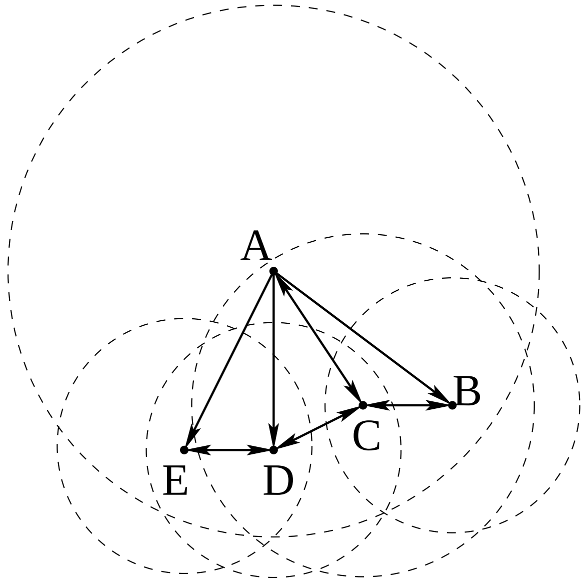

Figure 1: A unidirectional ad hoc network. A→B is a unidirectional link, and B→C→A is its reverse route.

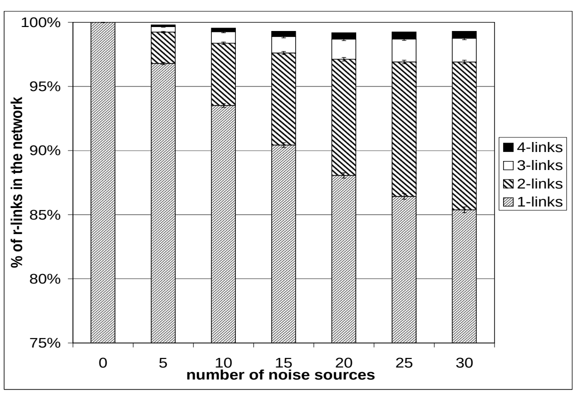

Figure 2(a): Average distribution of r-links vs number of noise sources for a density of 50 nodes/km^2. This figure shows that the number of r-links in the network decreases with r.

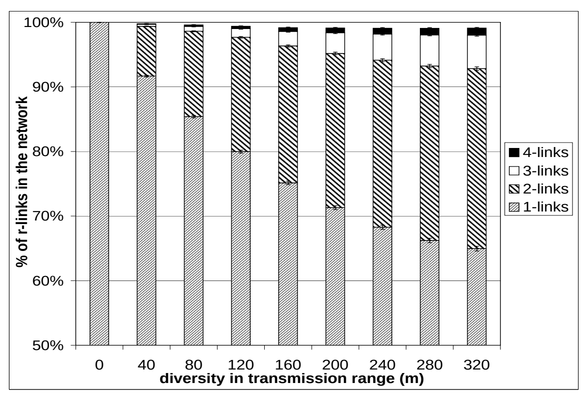

Figure 2(b): Average distribution of r-links vs diversity in transmission range for a density of 50 nodes/km^2. This figure shows that the number of r-links in the network decreases with r.

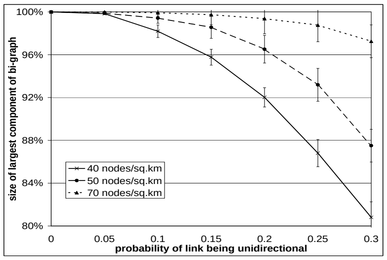

Figure 3(a): Normalized size of the largest connected component in the 1-graph vs probability of unidirectional links for different values of density. This figure shows that random unidirectional links can affect network connectivity significantly.

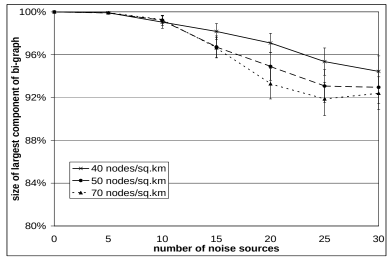

Figure 3(b): Normalized size of the largest connected component in the 1-graph vs number of noise sources for different values of density. This figure shows that unidirectional links due to noise can affect network connectivity significantly.

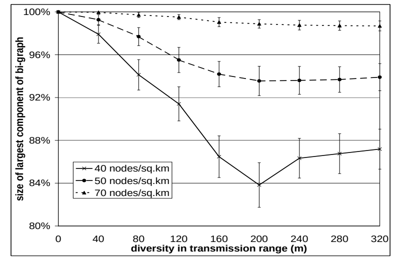

Figure 3(c): Normalized size of the largest connected component in the 1-graph vs diversity in transmission range for different values of density. This figure shows that unidirectional links due to diversity can affect network connectivity significantly.

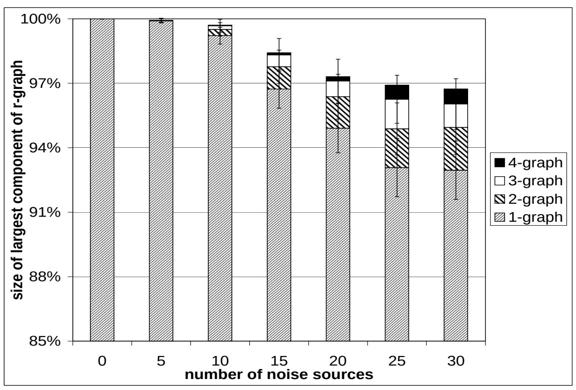

Figure 4(a): Normalized size of the largest connected component in r-graphs vs number of noise sources for a density of 50 nodes/km^2. This figure shows that unidirectional links due to noise can affect network connectivity.

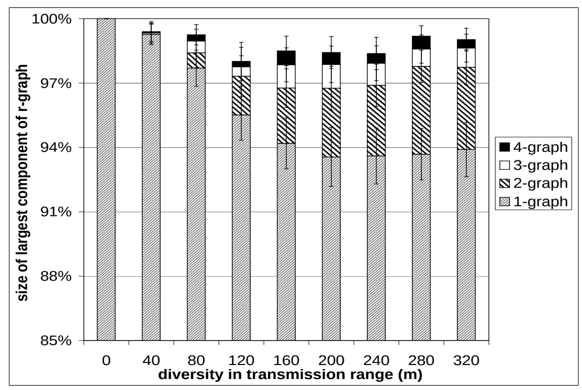

Figure 4(b): Normalized size of the largest connected component in r-graphs vs diversity in transmission range for a density of 50 nodes/km^2. This figure shows that unidirectional links due to diversity can affect network connectivity.

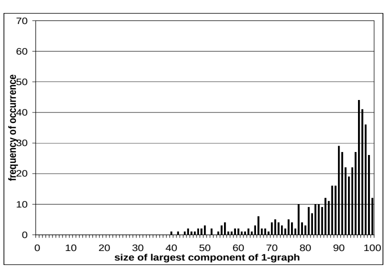

Figure 5(a): Histogram of the size of the largest component in the 1-graph for diversity 200m and density 50 nodes/km^2. This graph shows the heavy-tail distribution of connectivity.

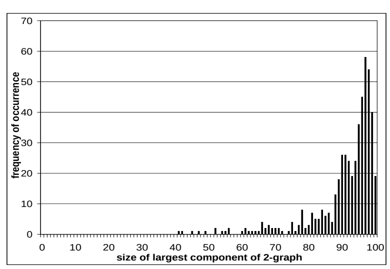

Figure 5(b): Histogram of the size of the largest component in the 2-graph for diversity 200m and density 50 nodes/km^2. This graph shows the heavy-tail distribution of connectivity.

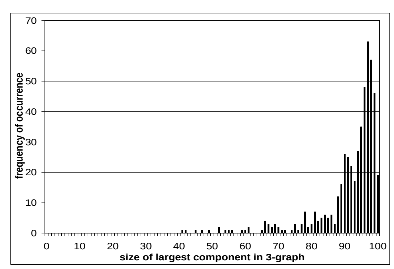

Figure 5(c): Histogram of the size of the largest component in the 3-graph for diversity 200m and density 50 nodes/km^2. This graph shows the heavy-tail distribution of connectivity.

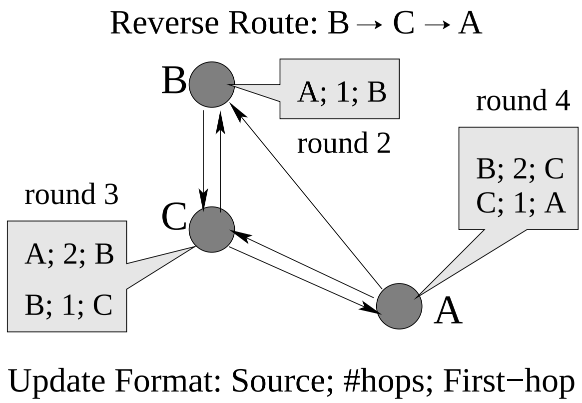

Figure 6: Reverse Distributed Bellman-Ford Algorithm. This figure illustrates how updates are propagated in RDBFA, enabling nodes to discover reverse routes. In this example, node B discovers the reverse route B→C→A of unidirectional link A→B.

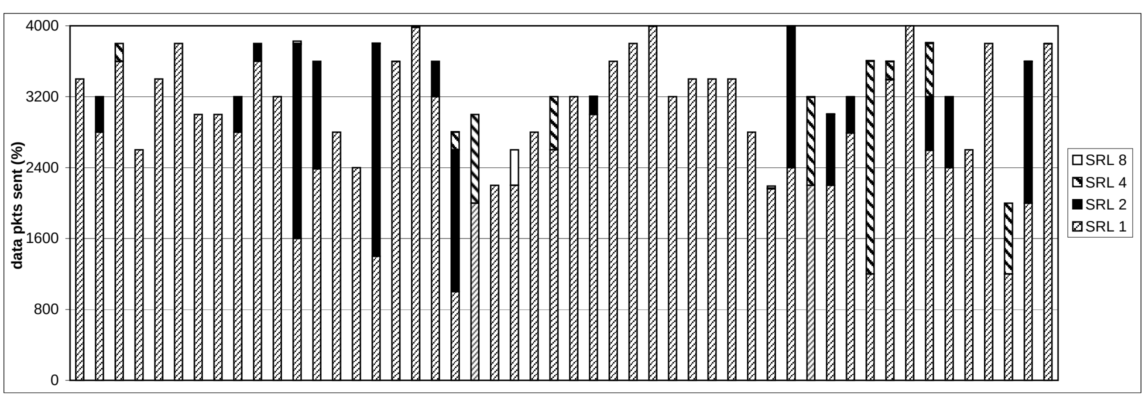

Figure 7: Number of data packets sent in different trials for diversity 280m and no mobility. This figure shows that in several trials, SRL significantly improves connectivity using unidirectional links.

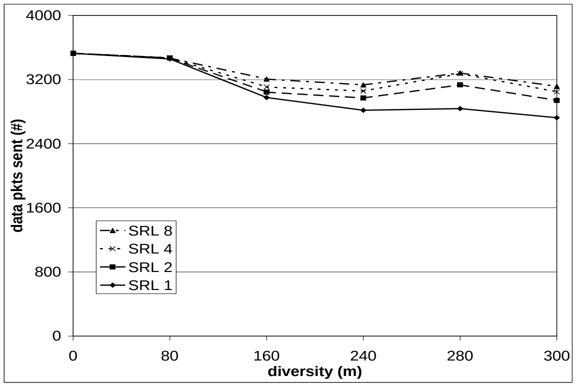

Figure 8(a): Average number of data packets sent vs diversity when all nodes are stationary. This figure shows that SRL uses unidirectional links to send more data packets.

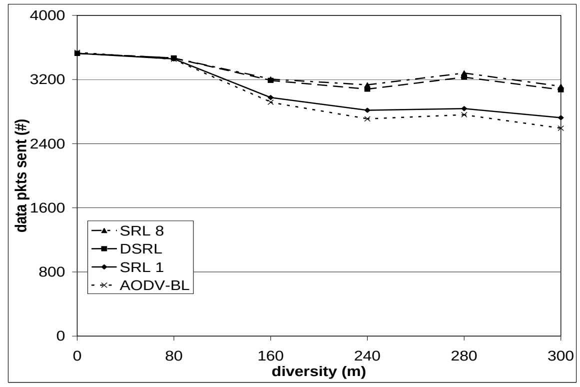

Figure 8(b): Average number of data packets sent vs diversity when all nodes are stationary. This figure shows that SRL and DSRL send significantly more packets than AODV-BL.

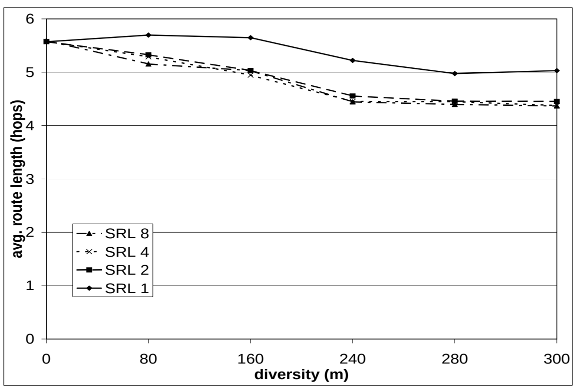

Figure 8(c): Average route length vs diversity when all nodes are stationary. This figure shows that SRL finds shorter routes by leveraging unidirectional links.

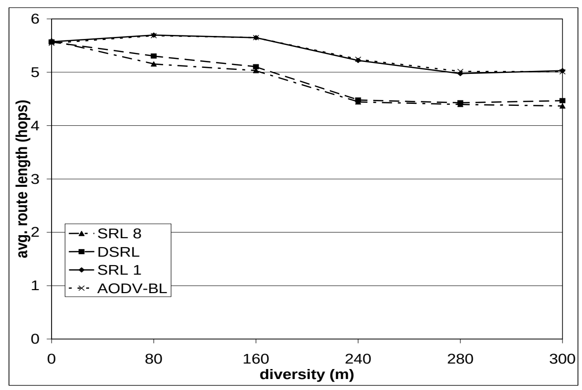

Figure 8(d): Average route length vs diversity when all nodes are stationary. TThis figure shows that SRL and DSRL find shorter routes than AODV-BL.

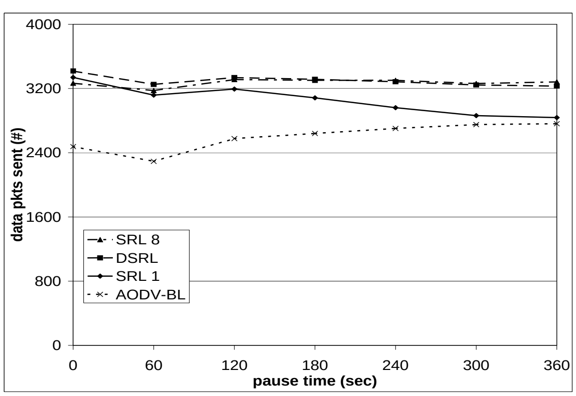

Figure 8(e): Average number of data packets sent vs pause time for diversity 280m and speed 10 to 12 m/s. This figure shows that SRL and DSRL send significantly more packets than AODV-BL.

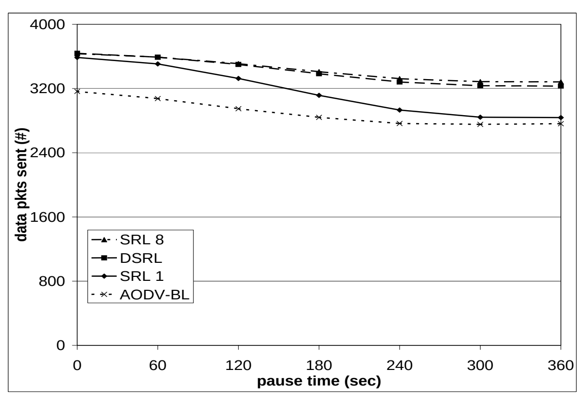

Figure 8(f): Average number of data packets sent vs pause time for diversity 280m and speed 1 to 2 m/s. This figure shows that SRL and DSRL send significantly more packets than AODV-BL.

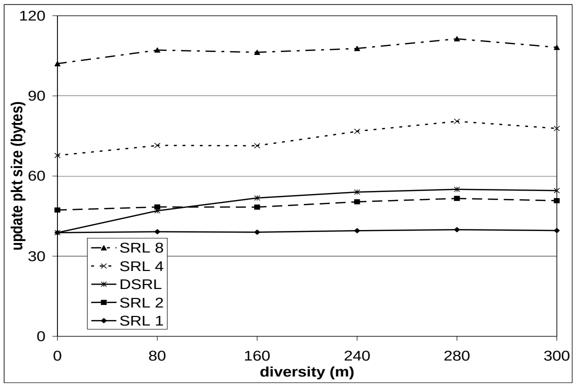

Figure 9(a): Average size of update packets vs diversity when all nodes are stationary. This figure shows that the overhead of DSRL is low and grows slowly with diversity.

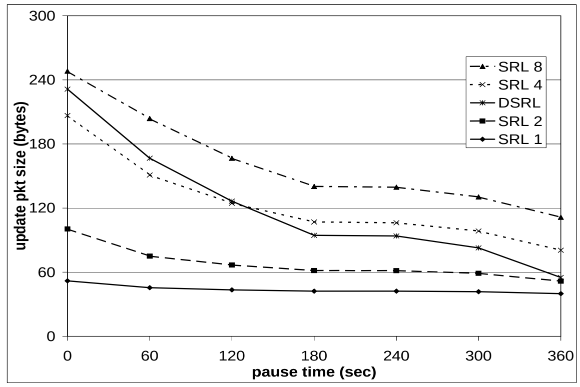

Figure 9(b): Average size of update packets vs pause time for diversity 280m and speed 10 to 12 m/s. This figure shows that the overhead of DSRL is low and grows slowly as mobility increases.

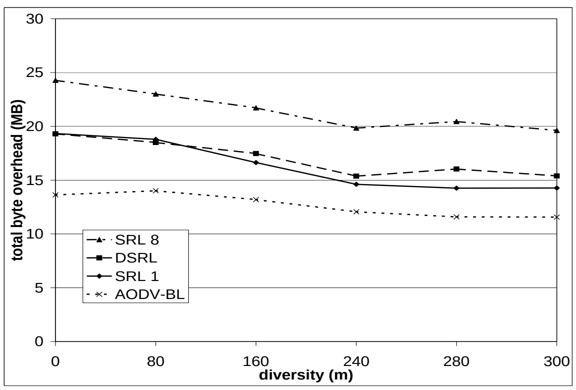

Figure 9(c): Total byte overhead vs diversity when all nodes are stationary. This figure shows that the overhead of DSRL is low and grows slowly with diversity.

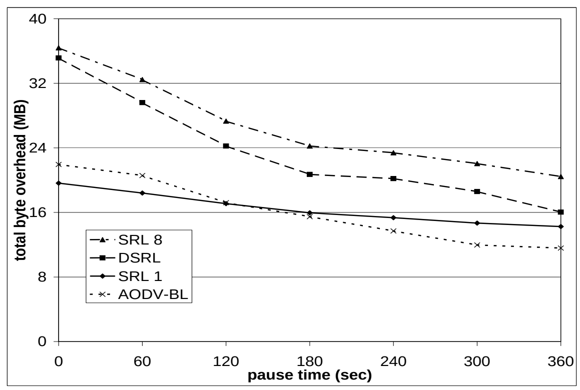

Figure 9(d): Total byte overhead vs pause time for diversity 280m and speed 10 to 12 m/s. This figure shows that the overhead of DSRL is low and grows slowly as mobility increases.

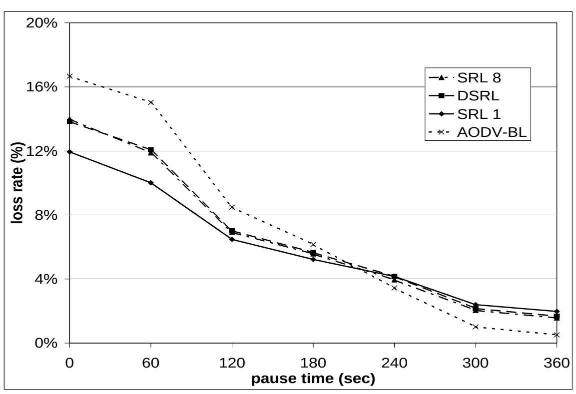

Figure 10: Average loss rate vs pause time for diversity 280m and speed 10 to 12 m/s. This figure shows that the loss rate of SRL and DSRL is comparable to or better than AODV-BL.

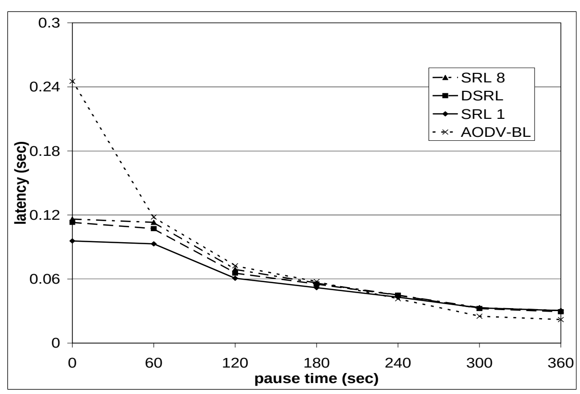

Figure 11: Average latency vs pause time for diversity 280m and speed 10 to 12 m/s. This figure shows that the latency of SRL and DSRL is as good as or better than AODV-BL.